How to / Trackmania / Manialink / Building a Maniapage

Author : Marcel (Translations: JonTheKiller, Marcel)

LAST PAGE PAGE 1 2 3 4 NEXT PAGE

The new elements of Forever ManiaLinks

<frame>

A frame has the ability to **group several elements** of a characteristic area (e.g. the header) to bring a kind of structure in the ManiaLink code. To improve this structure, it is possible to interlace the frames.

The advantage of using frames is that all included tags orientate on its position: If you move the frame you will also move all included elements into the same direction. Consequently it is an easy way to move your navigation around while developing your design ;)

To make it clear: The frame creates a new point of origin on its own positioning point. The included elements orientate on this new point of origin, not on the center of screen ;)

Attributes:

* pos(n)="X Y Z" – Position of the frame

Example:

```
<frame posn="20 20 0">
<quad posn="10 -10 0" />
</frame>
```

This frame orientates itself in the top right of the screen center. All tags included orientate themselves on the point (20, 20, 0) as the new point of origin. If you want to take the quad out of the frame and have it to be on the same position as before you have to change ist coordinates from (10, -10, 0) to (30, 10, 0). Of course you can place more tags in the frame, but since simplify matters only one is shown in the example.

<quad>

The quad is one of the new important elements in Forever ManiaLinks. It is used to place images on the ManiaLink and to react on the onMouseOver event (showing another image if the mouse is moved into the quad).

The quad is a "stand-alone" tag meaning that nothing is defined between <quad>-start and </quad>-end tag. For this XML has a shortform, write <quad ... /> instead of <quad ...></quad>. This works with all stand-alone tags.

Attributes:

* **pos(n)="X Y Z"** – Position of the quad

* **size(n)="width height"** – Size of the quad

* **scale="factor"** - Scale factor for the quad

With the scale factor you can define the real dimension of the quad. A value of "1" means the original size as given with size(n). If you write e.g. scale="2", the quad will displayed with double size in the end, as specified with size, a value smaller than 1 will reduce the size.

* **halign="left|center|right"** – Horizontal alignment

* **valign="top|center|bottom"** – Vertical alignment

* **image="image.jpg"** – Image URL
As mentioned a quad is mainly used to display images. The image is adjusted to the quad size, it fills out the quad completely. Additional to the image formats allready supported in TMU .jpg, .tga and .dds TMUF supports .gif (256 colors, with a transparent color) and .png (24bit, with and without alpha channel).

* **imagefocus="image.jpg"** – Image URL for onMouseOver image
If the mouse is moved into the quad this image will be displayed instead of the image declared with the image-attribute. The other way round changes the displayed image back to the first one.
This only happens if you put a link on the quad, otherwise nothing will happen on onMouseOver.

* **style="CategoryName"** – Category name of the predefinied styles

* **substyle="StyleName"** – Name of the predefinied style
With this two attributes style and substyle you can use the predefined styles included in TMF, like menu elements or buttons. Both attributes are needed to achieve one style. Remember that you cannot use style/substyle and image/imagefocus together, since they are responsible for the same feature (displaying images) and lock up each other.
Smurf made a manialink on which all predifende styles for quads and labels are listed, see: Example

* **bgcolor="RGBA"** – Uniform background
With a quad you can display a uniform colored background. The value is conform to a 4 digit hexadecimal notation, like the textcolor attribute of the <format>-tag. The first 3 digits stand for the intensity of the colors: red, green and blue, whereas the last one indicates the transparency. Each digit must be in the range of 0..9 and A..F (like color formating in you nickname or in the chat with the difference the the transparency cannot be changed there).

This attribute can also be used in the <format>-tag but it only affects <quad>-tags, others cannot be colored :(

* **url="externalHomepage.html"** – Link to a external page

* **manialink="ManiaLink"** – Link to a Manialink (registered code or URL)

* **maniazones="Maniazones-Link"** – Link to ManiaZones
With these 3 attributes you can create a link, either to an external webpage (TMF minimizes and opens a browser window with the declared URL), to a ManiaLink or to ManiaZones (nothing known about that). Logically you only can use one of these 3 attributes ;)

* **addplayerid="1"** – Creates an identification link
This attribut, set on "1", makes an identification link out of the link, that means you can access the data of the player added to the link. This can be used with PHP, following data is added:
playerlogin=m4rcel – Account login

lang=de – Language of Trackmania Forever

nickname=FT»Marcel – Nick name inclusive formating (special characters forged, e.g. » becomes %bb)

path=World/Germany/Thuringia/Erfurt – Location specified in the profil (/ becomes %7c)
With this information you can identify the player to give him access to special sections on your ManiaLink.

A link would look like this:
page.php?playerlogin=m4rcel&lang=de&nickname=FT%bbMarcel&path=World%7cGermany%7cThuringia%7cErfurt
The "page.php" is the url specified with **url**, **manialink** or **maniazones**.

Important hint: Unfortunately AddPlayerID is not as save as it seams. Indeed an identification of the player is possible, but this can easily be manipulated by entering the URL manuallay. Do not use it for critical sections!

Example:

```
<quad
pos="-0.5 0.25 -0.5" size="1.2 0.2"
valign="center"
image="http://funtrackers.bplaced.net/images/tm_island/header.png"
imagefocus="http://funtrackers.bplaced.net/images/tm_desert/header.png"
manialink="http://localhost/manialink/index.xml"
/>
```

This quad has the island header of the FT-forum as background. On *onMouseOver* it changes to the desert header. The file links to itself, which is placed on my local server, so I have not to type in the URL again and again :P The quad is placed top right of the screen center. With Z=-0.5 it is in the foreground.

```
<quad
posn="0 0 0"
sizen="32 32"
halign="left"
valign="top"
style="Bgs1InRace"
substyle="BgWindow1"
/>
```

This is the standart quad from the quad-positioning-script. Here you see the usage of the predefinded styles. The combination of "Bgs1InRace" and "BgWindow" creates an rather dark rectangle with round edges.

<format>
The format tag as an non-visuall element groups several formattings which are used by different other elements. The format is used for the whole manialink or, if the format tag is placed in a frame, in frame and his subframes only.
Affected by this tag are the <label>, <entry> und <fileentry> tags, but they always can overwrite predefined formats with the same attributes.

Attributes:
* **textsize="Number"** – Text size
* **textcolor="RGBA"** – Text color
Uses same color system like the <quad>-tag for the background.

* **style="StyleName"** – Name of the predefinied style

Please note that only styles of labels can be used here, they differ from that ones of quads. What styles exist you can see on Smurfs manialink Example for styles.

Some styles, like CardButtons, affect the size of the label.

Example:

```
<format style="TextValueSmall" />
```

With this all <label>, <entry> and <fileentry> tags get the TextValueSmall-format. In the same way you could set the text size and color.

<label>
The label is one of the new important elements. It is used to display text on the ManiaLink and overtakes the task of the old Text element. This means even longer texts are displayed using a label on Forever ManiaLinks.
Since the label is one of the elements which can make use of the format feature you can easily format your text style.
Another feature is the autotranslation of certain words within the label. Because of this the word "Statistics" is automatically translated depending the used language, e.g. in German into "Statistik". This only works with stand-alone-words, if there is only one character more it will not work. Which is quite usefull if it should not be translated on its own: only add a space.

Attributes:

* **pos(n)="X Y Z"** – Position of the label

* **size(n)="width height"** – Size of the label

* **scale="factor"** - The scale factor for the label

* **halign="left|center|right"** – Horizontal alignment

* **valign="top|center|bottom"** – Vertical alignment

Both alignments influence the displayment of the text directly, it overtooks the alignments. The text oriantes as close to the positoning point as possible. So if the label has the settings bottom and center for the alignments the text will be centered at the bottom of the label. Regarding multiple line texts not each row is aligned on its own. All lines are positionated on one line (concerning *halign*).

* **text="LabelText"** – The text of the label
With this attribute you indicate the displayed text.
Word-wrap within the text will cause a word-wrap on the manialink.
Instead of <label text="LabelText" /> you can also write: <label>LabelText</label> (which looks like the old text element).
The text can include all $-formatings.

* **autonewline="1"** – Activates auto word-wrap
With this attribuet you can activate an auto word-wrap. With that to long lines are split up into more lines.
Otherwiese the text will be displayed in one line and, if the text is too long, will be compress to the width of the label.

* **url="externalHomepage.html"** – Link to a external page

* **manialink="ManiaLink"** – Link to a Manialink (registered code or URL)

* **maniazones="Maniazones-Link"** – Link to ManiaZones

If no style is set the whole area of the label will become the link and not only the text. If the link is created the attribute autonewline will not work (a bug I allready reported in the beta)

* **addplayerid="1"** – Creates an identification link
Additional information see attribute addplayerid in the Quad description

* **<*format*>** - Attributes

Example:

```
<label posn="-30 -20 5" halign="center" style="CardButtonMedium" text="$o$008White Button with darkblue Font" />
```

This label is displayed in the bottom left corner and has next to the predefined style some $-formats.

<entry>

A key innovation of the new ManiaLinks is the ability to read the player's input and to process it in PHP-Scripts. You can use the "Entry" element for the input, which automatically inserts its value in the URL, where its name appears. As Quad the Entry element is a stand-alone tag, and contains no text between the tags.

An Entry accepts always strings, a possible restriction of the values must be done with the script, which could put out an error message for invalid values (e.g. if the value was not numeric).

It should be self-explanatory that an Entry element is only useful in a conjunction with a dynamic (script-based) ManiaLink. In the opposite a static (XML-based) ManiaLink is not able to respond to any input ;)

Attributes:

* **pos(n)="X Y Z"** - The position of the Entry

* **size(n)="Width Height"** - The size of the Entry

* **scale="factor"** - The scale factor for the Entry

* **halign="left|center|right"** - Horizontal alignment

* **valign="top|center|bottom"** - Vertical alignment

* **name="EntryName"** - The (unique) Name of the Entry
In order to respond an input, the name of the Entry is important: If in any link this name occurs, it will be automatically replaced by the current value of this Entry. That is the reason why it is important that all Entries have unique names.

* **default="value"** - The default value of the Entry
This value is displayed in the Entry when you start the ManiaLinks. As long as the player does not change it.

* **autonewline="1"** – Enables the wordwrap
AutoNewLine can be used for larger Entries. It is useful for each input which is bigger than only one line. It could be set in the Source with AutoNewLine. Without this claim the file will always show you the first line, also when you turn the enter key.

* **<*format*>** attributes

Example:

```
<entry posn="15 -8 5" sizen="5 2" style="TextValueSmall" name="inputvalue" default="42" />
<label posn="15 -1 -12 5" style="CardButtonMedium" text="Link with value" url="http://site.php?value=inputvalue" />
```

This Entry shows the value 42 when you open the ManiaLink. This value could be changed by the player at any time. If the player clicks on the label (which acts as a displayed button), "inputvalue" will automatically replaced by his command, and finally evaluated in the site.php.

<fileentry>

It is like the name says: The FileEntry has the task to receive complete files and send them to a working script. If this element is clicked, a little window opens for the player. The player can choose in the window his file for upload. This action requires good knowledge of PHP, because finally the file must be received (POST method) and saved by the server. (At this point, my knowledge ends. I am not able to say anything more about it ^^)

Attributes:

* **pos(n)="X Y Z"** – The position of the FileEntry

* **size(n)="Width Height"** – The size of the FileEntry

* **scale="factor"** - The scale factor for the FileEntry

* **halign="left|center|right"** – The horizontal alignment

* **valign="top|center|bottom"** – The vertical alignment

* **name="FileEntryName"** – The name of the FileEntry
The name of the FileEntry has the same function like the name of the entry.
For more information please look at the entry part of this tutorial.

* **folder="folderpath"** – The path of the folder at the top.

The folder which you put in the value folder will be shown at first. It is also the highest folder which is available. The default folder is the TrackMania folder in your documents.

Important hint: Without the value folder your TMF will crash! If you want to choose the default folder of TMF (the folder in your documents), use folder="" ;)

* **autonewline="1"** – Enables the wordwrap
For more information for this value please read the entry part of this tutorial.

* **default="value"** – The standard value of FileEntry
The default value of FileEntry has the same task as in Entry: This value is shown when opening the ManiaLink, and will be replaced after the player has chosen a file.

* **<*format*>** -attributes

Example:

```
<fileentry sizen="39 2" posn="16 -28 0" halign="left" style="TextValueSmall" name="avatarfilename"
folder="skins\avatars" default="??"/>
<label posn="1 -32 0" halign="left" style="CardButtonMediumWide"
manialink="POST(http://localhost/ManialinkSamples/UploadAvatar.php?file=avatarfilename,avatarfilename)"
addplayerid="1" text="Post with playerid, target=manialink"/>
```

This code was taken from the ManiaLink example of Nadeo, which is available since Beta phase. This specified FileEntry starts in the "skins\avatars" folder, so that the player is only able to upload an avatar or any other file, which was previously saved in his avatar folder (local). The label shows you how the file will be send to a suitable script: With the help of POST() the file can be received by the script and saved on the server. (The AddPlayerID attribute is needed to generate a unique name for this file)

<timeout>
In TMU you had the problem, that ManiaLinks were cached, that means they were saved locally and on next call of this ManiaLink only the local copy was loaded instead of requesting it from the server again. In relation of dynamic ManiaLinks where content changes in rapid succesion this caching was a real problem. With help of header manipulations you could avoid this caching, but this way was not really nice ^^

Now you have the TimeOut tag: You have only to specify the time interval (in seconds) after which the ManiaLink should be reloaded from the server instead of using the cached copy of it. Using 0 as time interval means, that TMF will request the file from the server on each call of the ManiaLink.

Example:

```
<timeout>0</timeout>
```

The two tags enclose a zero. Any questions? :D

With this piece of code you force TMF to reload the XML or PHP file from the server, so that possible changes will be applied with the next call of the ManiaLink without having to restart TMF. Use this code, if your ManiaLink is still under construction, so that you really can see all changes ;)

<include>
PHP coder already know the function of this tag: With help of Include you can import another XML file into the current one. This means that often used contents can be excluded into seperate files and in the special ManiaLinks you only have to insert the Include tag.

The file which should be included only needs the ManiaLink elements, without XML header or the <manialink>. See the including procedure as following: TMF cuts the XML file into two parts, and inserts the code of the external XML file 1:1 into this gap. So an additional <manialink> would be missplaced ;)

But there is a little bug, which I noticed already in the Beta phase: Only the first element of the external file is imported, all following elements are ignored by TMF. A little workaround is, to take all elements into an own frame, so that the frame is selected to be imported, and with it alle the other elements.

Disadvantage: Because the music tag is not working in frames, this tag can only be imported alone.

Attributes:

* **url="inc.xml"** – File to be imported

Example:

ManiaLink file:

```
<?xml version="1.0" encoding="utf-8" ?>
<manialink>
<include url="http://localhost/manialink/inc.xml"/>
</manialink>
```

inc.xml:

```
<frame>
<quad posn="-20 -20 0" sizen="20 20" style="Bgs1InRace" substyle="BgWindow2" />
<quad posn="20 20 1" sizen="20 20" style="Bgs1InRace" substyle="BgWindow1" />
</frame>
```

The main file only has the include element. To import both of the quads, you have to put them into a frame. Otherwise only the first quad would be visible, and the second one would be ignored.

<music>

Up to now, you had to use the audio element for playing some music in the background, which size was set to 0*0 for hidding the start/stop button. For this case there now is the <music> tag. After specifying the music file, which has to be in .ogg or .mux format, the music will be played without any visual displaying. A nice secondary effect: If you change the ManiaLink and the following ManiaLink has the same background music, it will go on without any break ;)

The music tag does only work, if it has the <manialink> directly as parent. If it is placed in a frame, you will hear the default menu music.

The best way is, to take the ManiaLink element to the end of the XML file, so the whole ManiaLink has been loaded before requesting the music file, which could be a bit bigger ;)

Attributes:

* **data="music.ogg"** – The music file

Example:

```
<music data="http://localhost/manialink/alpine_race.ogg"/>
```

Simple function, simple code. You only have to specifiy the URL (here it is the Snow music from TMO) and it's done. (Because simple is simply simple :D)

<audio>

One of the few elements, which work completely in the new structure of the Forever-ManiaLinks, is the audio tag. As known from TMU, you can play a music file with the audio. The difference betwenn <music> and <audio> is that the audio tag has a Play and Stop button, so the user can choose if he wants to hear it or not. The valid formats for <audio> are .ogg and .mux.

Attributes:

* **pos(n)="X Y Z"** – The postion of the Audio

* **size(n)="Width Height"** – The size of the Audio
Because of the downward compatibility to TMU, you can choose to use width="Width" and height="Height" instead (Both only in classic positioning).

* **scale="factor"** - The scale factor for the audio

* **data="audio.ogg"** – The audio file
You can write the filename between the <audio>-tags as in TMU instead of using the attribute data.

* **play="1"** – Automatic start up when opening ManiaLink
If the attribute play is set to "1", the audio file will start playing when the ManiaLink is opened.

* **looping="0"** - Disable looping of the audio
If looping is set to "0", the audio file will end after it played once. If set to "1" (default), it will play in an infinity loop.

Example:

```
<audio pos="0 0 0" data="http://localhost/manialink/alpine_race.ogg"/>
```

The start button, displayed in the middle, will start playing the TMO Snow music when clicked

<video>

The video element is supported completely, too. It works almost like the audio element, only that it is displayed in another way (it is a video and not a music file ^^). The valid format for the video element is .bik.

Attributes:

* **pos(n)="X Y Z"** – The position of the video

* **size(n)="Width Height"** – The size of the video on the screen

As in the audio tag, there are the two old elements for using: width="Width" and height="Height" (Both only in classic positioning)

* **scale="factor"** - The scale factor for the video

* **data="video.bik"** – The video file
Instead of using the data attribute, you can write the filename between the <video>-tags, as it was in TMU.

* **play="1"** – Automatic start on opening the ManiaLink
If the attribute play is set to "1", the video will also be played on opening the ManiaLink.

* **looping="0"** - Disable looping of the video
If looping is set to "0", the video will end after playing once, as it is with audio. Set to "1" (default) means repeating the video in an infinity loop.

Example:

```
<video pos="0 0 0" size="0.512 0.256" data="http://localhost/manialink/sign_warning.bik" play="1"/>
```

With this code the SignWarning sign will be automatically begin to play when opening the ManiaLink.

This site is friendly to stationary (PC/MAC) and mobile (Android/IOS) browsers
Trackmania, Shootmania, Questmania games & media © Nadeo & UbiSoft used under fair use policy
8,373,157 page views since January 2008. All tutorials, services, coding © Mania Creative 2008-2013